Claims:

1. A processor having a real time debugging interface, said processor comprising:

a) instruction memory means for storing instructions to be executed by said processor;

b) program counter means ~~coupled~~ *directly coupled connected* to said instruction memory means for indexing said instructions;

c) cause register means for indicating information regarding interrupts and exceptions; and

d) first decoder means for indicating information about an instruction executed by said processor during a clock cycle, said first decoder means being ~~coupled~~ *directly coupled connected* to said instruction memory means, said program counter means, and said cause register means, said first decoder means having a first output, wherein

said first output provides information regarding activity of said processor in real time.

2. A processor according to claim 1, said information regarding processor activity includes information as to at least one of a jump instruction has been executed, a jump instruction based on the contents of a register has been executed, a branch has been taken, and an exception has been encountered.

25

3. A processor according to claim 1, wherein:

said clock cycle is a processor clock cycle, and

said first decoder means updates said information about each

instruction executed by said processor for each said processor

clock cycle.

4. A processor according to claim 3, wherein:

said information about each instruction executed by said

processor includes an indication whether or not an instruction has

been executed since *the* previous processor cycle.

5. A processor according to claim 1, wherein:

said first output *consists of* a three bit parallel output.

26

6. A processor according to claim 1, further comprising:

e) second decoder means *directly coupled* ~~connected~~ ~~coupled~~ to said cause register means for indicating information about contents of said cause register means, said second decoder means having a second output; and

f) event history buffer means for storing information regarding processor events, said event history buffer means having a data input, a data output, and an enable input, said data input being ~~connected~~ *directly coupled* ~~coupled~~ to said cause register means and said enable input being ~~connected~~ *directly coupled* ~~coupled~~ to said second output, wherein

said second decoder means decodes contents of said cause register means and enables said event history buffer means to capture contents of said cause register means when contents of said cause register means indicate a particular event.

7. A processor according to claim 6, wherein:

said second decoder means enables said event history buffer means when contents of said cause register means indicate an event including at least one of a change in status of an interrupt line, an internal processor exception, and a jump instruction based on the contents of a register.

8. A processor according to claim 6, wherein:

said data output of said event history buffer means is a bit serial output.

27

9.   A processor according to claim 6, wherein:

said processor is embodied on a chip having a plurality of pins,

said first output and said data output are provided via some of said plurality of pins.

10.   A processor according to claim 9, wherein:

said first output is an n-bit parallel output, and

said data output is a serial output.

11. An embedded system having a plurality of processors and a real time debugging interface, said system comprising:

a) a plurality of instruction memory means for storing instructions to be executed by a respective one of said plurality of processors;

b) a plurality of program counter means, each ~~coupled~~ *directly coupled* ~~connected~~ to a respective one of said plurality of instruction memory means for indexing contents of said instruction memory means;

c) a plurality of cause register means for indicating information regarding interrupts and exceptions for a corresponding one of said plurality of processors, each of said cause register means being ~~coupled~~ *directly coupled* ~~connected~~ to a respective one of said processors; and

d) a plurality of first decoder means, each said first decoder means *directly* ~~connected coupled~~ ~~coupled~~ to a respective one of said instruction memory means, to a respective one of said program counter means, and a respective one of said cause register means, each said first decoder means for indicating information about an instruction executed during a clock cycle by a respective one of said processors, each said first decoder means having a first output, wherein

each said first output provides information regarding activity of said processor in real time.

12. An embedded system according to claim 11, wherein:

said information regarding processor activity includes information as to at least one of a jump instruction has been executed, a jump instruction based on the contents of a register has been executed, a branch has been taken, and an exception has been encountered.

13. An embedded system according to claim 11, wherein:

said clock cycle is a processor clock cycle, and

each said first decoder means updates said information about each instruction executed by a respective processor for each said processor clock cycle of said respective processor.

14. An embedded system according to claim 13, wherein:

each said information about each instruction executed by a respective processor includes an indication whether or not an instruction has been executed since the previous processor cycle of said respective processor.

15. An embedded system according to claim 11, wherein:
each of said first outputs Consists of a three bit parallel output.

30

- 28 -

16. An embedded system according to claim 11, further comprising:

e) a plurality of second decoder means, each ~~coupled~~ directly coupled to a respective one of said plurality of cause register means, each said second decoder means for indicating information about contents of a respective cause register means; and

f) an event history buffer means for storing information regarding processor events, said history buffer means having a data input, a data output, and an enable input, said data input being ~~coupled~~ directly coupled to each of said plurality of cause register means and said enable input being ~~coupled~~ directly coupled to each of said second outputs, wherein

each of said second decoder means decodes contents of a respective cause register means and enables said event history buffer to capture contents of said respective cause register means when contents of said respective cause register means indicate a particular event.

17. An embedded system according to claim 16, wherein:

each said second decoder means enables said event history buffer means when contents of a respective cause register means indicate an event including at least one of a change in status of an interrupt line, an internal processor exception, and a jump instruction based on the contents of a register.

31

18. An embedded system according to claim 16, wherein:

said data output of said event history buffer means is a bit serial output.

19. An embedded system according to claim 11, wherein:

said system is embodied on a chip having a plurality of pins,

said first and second outputs are provided via some of said plurality of pins.

20. An embedded system according to claim 19, wherein:

each of said first outputs is an n-bit parallel output, and

said second output is a serial output.

21. ~~A method of debugging a processor, said method comprising:~~

a) providing information about processor activity in real time; and

b) associating the instructions executed by the processor with the information about processor activity.

22. A method according to claim 21, wherein:
said step of providing information about processor activity
includes providing information about every instruction executed by
the processor.

23. A method according to claim 22, wherein:

said step of providing information about processor activity

includes providing information that the processor has not executed

an instruction during the last processor cycle.

24. A method according to claim 21, wherein:

the information about processor activity includes an

indication of at least one of whether the last instruction

executed was a jump, a jump based on the contents of a register, a

branch taken, or an instruction which encountered an exception.

25. A method according to claim 21, further comprising:

c) providing information regarding the status of the processor

when certain processor events occur, said certain processor events

including at least one of a change in status of an interrupt line,

an internal processor exception, and the execution of a jump

instruction based on the contents of a register.

- 31 -